Inteligência Artificial para Apoio à Escrita Coletiva Digital

CASTRO, Isabella de Medeiros
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
isabella-medeiros@hotmail.com

GUIZZO, Michele Alda Rosso
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
michele.guizzo@ifsc.edu.br

Resumo — Este artigo apresenta o desenvolvimento, a integração e a avaliação de uma ferramenta de consulta inteligente baseada na técnica Retrieval-Augmented Generation (RAG), incorporada ao Editor de Texto Coletivo (ETC), plataforma de escrita colaborativa desenvolvida pela UFRGS. O sistema permite que usuários consultem documentos previamente indexados por meio de perguntas em linguagem natural, com entrada por texto ou voz. A solução utiliza exclusivamente ferramentas e modelos de código aberto, incluindo o modelo Mistral-7B-Instruct, o mecanismo de recuperação vetorial FAISS e o modelo Whisper para reconhecimento de fala. A metodologia envolveu uma revisão sistemática de literatura, a análise de ferramentas, o desenvolvimento da arquitetura, a integração com o ETC e testes funcionais. Na avaliação de desempenho, nove perguntas foram aplicadas ao sistema, divididas em três categorias — factuais, inferenciais e multi-hop — e avaliadas pelas métricas Precision@3, ROUGE-L, acurácia e latência. Os resultados indicaram acurácia de 77,7%, com melhor desempenho em perguntas factuais e desempenho reduzido nas questões inferenciais devido a limitações no processo de recuperação semântica. A análise demonstra que o componente mais crítico do sistema é a etapa de recuperação, sugerindo melhorias como redução do tamanho dos *chunks* e otimização dos embeddings. A ferramenta integrada ao ETC mostrou-se funcional, contribuindo para o apoio à escrita coletiva e indicando potencial para futuras expansões, como interpretação de textos presentes em imagens via OCR.

Palavras-Chave — Inteligência Artificial; RAG; Recuperação Semântica; Modelos de Linguagem; Escrita Coletiva.

I. INTRODUÇÃO

A Inteligência Artificial (IA) tem se consolidado como uma das áreas mais dinâmicas da ciência da computação. No campo educacional, as tecnologias baseadas em IA têm possibilitado o desenvolvimento de ferramentas capazes de acompanhar o desempenho dos estudantes em tempo real, automatizar tarefas administrativas e apoiar práticas pedagógicas por meio da análise de dados. Assim, a integração da IA à educação vem se mostrando promissora, especialmente no apoio à escrita, à leitura e à construção de conhecimento [1].

Os LLMs (*Large Language Models*) são modelos de IA treinados em grandes volumes de texto para aprender a entender e gerar linguagem humana, utilizando redes neurais e algoritmos de aprendizado de máquina (*Machine Learning*). Dessa forma, o treinamento permite que os modelos reconheçam padrões linguísticos e gerem respostas e textos de forma precisa. Os modelos LLMs são treinados com dados de artigos, livros e postagens de mídia social, e esse processo exige recursos computacionais significativos [2]. Na educação, a Inteligência Artificial tem potencial para transformar o processo de ensino e aprendizagem, tornando-o mais adaptativo e eficiente.

O Editor de Texto Coletivo (ETC), é uma plataforma de escrita coletiva desenvolvida pela UFRGS (Universidade Federal do Rio Grande do Sul), que tem sido continuamente aprimorada visando contribuir para a qualificação da produção textual de forma dinâmica e interativa. Para isso, o ETC conta com diversas ferramentas que apoiam a construção de textos. Contudo, apesar dos avanços na plataforma, ainda há desafios no acesso e na organização das informações.

Nos ambientes acadêmicos e profissionais, encontrar rapidamente conteúdos relevantes em um grande volume de textos pode ser demorado e pouco eficiente. A busca tradicional por palavras-chave nem sempre retornam os trechos mais adequados ou fundamentados [3]. Depois disso, os estudantes precisam ainda selecionar trechos, organizar ideias e buscar entre as diferentes fontes recuperadas para construir seus argumentos. A ausência de mecanismos que colaborem com esse processo comprometem a fluidez da escrita.

Neste sentido, este trabalho irá explorar o uso da Inteligência Artificial para uma consulta de informações em bases de dados qualificadas pelo usuário. O objetivo é apresentar os resultados do desenvolvimento de uma ferramenta capaz de responder perguntas relacionadas aos textos selecionados pelo usuário. O sistema será implementado no Editor de Texto Coletivo (ETC), uma plataforma assíncrona de escrita coletiva. A nova funcionalidade utiliza recursos de IA e poderá ser operada tanto por comandos de texto quanto por comandos de voz.

A solução proposta visa amplia o Editor de Texto Coletivo ao incorporar uma funcionalidade de consulta ancorada em evidências textuais, permitindo que os estudantes dialoguem com suas fontes de forma mais sistemática e reduzam o tempo de busca de informações relevantes, mantendo o vínculo entre as respostas geradas e os documentos utilizados como base. Além disso, a ferramenta pode apoiar práticas pedagógicas como pesquisa guiada, análise de fontes e discussões colaborativas, nas quais os estudantes formulam perguntas, confrontam diferentes trechos textuais e constroem argumentos fundamentados.

Dessa forma, pretende-se apoiar o processo de produção textual dos usuários no ETC, possibilitando reflexões e discussões sobre os textos pré-selecionados, colaborando assim com a qualidade da produção textual dos estudantes.

O presente artigo está dividido em cinco seções. Inicialmente são apresentados os referenciais teóricos, depois a metodologia proposta pelo trabalho, e por fim, os resultados e discussões, e a conclusão.

II. REFERENCIAL TEÓRICO

A seção de Referencial Teórico apresenta os fundamentos de Inteligência Artificial e de suas principais vertentes, de modo a oferecer o embasamento teórico necessário para a compreensão das tecnologias utilizadas ao longo deste estudo. Além disso, descreve a ferramenta de edição de textos coletivos ETC e suas funcionalidades.

A. Inteligência Artificial Generativa - LLMs

No âmbito da linguagem, o Processamento de Linguagem Natural (*Natural Language Processing* – NLP) é o ramo da IA dedicado a permitir que os computadores compreendam, interpretem e gerem a linguagem humana. Nesse contexto, se inserem os Modelos de Linguagem de Grande Porte (*Large Language Models* - LLM) que aprendem a partir de grandes volumes de texto no pré-treinamento, adquirindo conhecimento sobre padrões de linguagem humana e sobre o mundo. Logo, isso os torna eficazes em tarefas de processamento de linguagem natural que envolvem geração de texto, como sumarização, tradução, perguntas e respostas e chatbots [4].

B. Retrieval-Augmented Generation - RAG

Em sistemas de Perguntas e Respostas (*Question Answering* - QA), uma abordagem tradicional para gerar respostas é a geração condicional simples, na qual modelo LLM gera a resposta palavra por palavra de maneira autoregressiva, ou seja, cada palavra é condicionada às anteriores. Além disso, esses sistemas são limitados ao contexto descrito no prompt e ao conhecimento pré-treinado do modelo. Assim, a abordagem faz com que aumente as chances de serem geradas respostas imprecisas e com ausência de evidências textuais fundamentadas [4].

Logo, para superar essas limitações, foi proposto o paradigma de Geração Aumentada por Recuperação (*Retrieval-Augmented Generation* - RAG). Neste método, antes da geração da resposta, são recuperados trechos relevantes de uma coleção de documentos, incorporados como contexto no prompt de um modelo LLM, com instruções do tipo: “Com base nesses textos, responda a esta pergunta:”. Assim, a geração autoregressiva é condicionada não apenas à pergunta, mas também aos documentos recuperados, oferecendo uma base factual para a resposta [4].

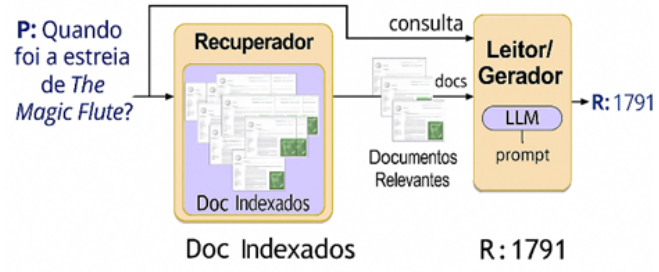

Figura 1: Esboço do modelo QA com RAG. Adaptado de [4].

Na Figura 1 é possível observar que, na primeira etapa, chamada de recuperador, os documentos indexados são divididos em *chunks* (trechos). Assim, apenas os trechos relevantes para a pergunta formulada no prompt são separados. Depois, na segunda fase de leitura e geração da resposta, os trechos recuperados (*retrieval*) são combinados à pergunta e enviados como entrada para o modelo LLM, que então gera a resposta com base nesse contexto.

C. Editor de Textos Coletivo (ETC)

O processo de escrita coletiva digital baseia-se em interações entre os participantes que constroem um texto em conjunto por meio de trocas, negociações e compartilhamento de objetivos comuns. Esse ambiente colaborativo favorece o uso de diferentes recursos, como artigos e reportagens, que contribuem para a produção textual e tornam o processo mais dinâmico [5]. O Editor de Texto Coletivo (ETC) é uma plataforma digital criada pelo Núcleo de Tecnologia Digital aplicada à Educação (NUTED) da Universidade Federal do Rio Grande do Sul, que vem sendo aprimorado com base nas experiências práticas dos alunos e professores (disponível em https://www.nuted.ufrgs.br/etc/). A Figura 2 mostra a interface de login da plataforma.

Figura 2: Interface de Login do ETC.

O sistema dispõe de ferramentas que favorecem a comunicação entre os participantes, como o envio de mensagens por e-mail, inserção de comentários diretamente no texto e fórum com visualizações cronológicas. Além disso, funcionalidades que ajudam na produção e qualificação do texto. As ferramentas colaboraram para a busca por referências, análise de conceitos e autoria e a consulta de emoções básicas dos estudantes durante a construção do texto. Todas essas funcionalidades foram implementadas a partir de projetos de pesquisa desenvolvidos pelo grupo.

III. METODOLOGIA

A metodologia adotada foi dividida em seis etapas, listados na Figura 3, e aborda desde a Revisão Sistemática

de Literatura até a Avaliação de Desempenho do sistema.

Figura 3: Fluxograma das Etapas da Metodologia.

A Revisão Sistemática de Literatura foi conduzida com o objetivo de identificar e analisar estudos relevantes sobre o uso da Inteligência Artificial na recuperação de informações em textos. O processo seguiu etapas metodológicas propostas por [6] e adaptadas por [7] que envolvem estratégias para busca, seleção e síntese de publicações acadêmicas relevantes.

A Análise e Seleção das Ferramentas foi realizada com base nos resultados obtidos na Revisão Sistemática de Literatura. Nessa etapa, avaliaram-se as tecnologias e os modelos de IA utilizados nos estudos analisados, com ênfase nas estratégias de recuperação da informação, bibliotecas empregadas e arquiteturas mais adequadas para integração com sistemas externos. Dessa forma, a análise orientou a escolha das ferramentas e técnicas mais apropriadas para o desenvolvimento da solução proposta.

A etapa de Desenvolvimento do Sistema adotou a técnica Retrieval-Augmented Generation (RAG), que combina recuperação semântica de trechos relevantes e geração de respostas por um modelo de linguagem. Essa abordagem permite que o sistema responda a perguntas fundamentadas exclusivamente nos documentos fornecidos pelo usuário, aumentando a precisão e reduzindo erros de alucinação. Além disso, foi incorporado um módulo de comando de voz, responsável por converter fala em texto.

A integração com o Editor de Texto Coletivo (ETC), plataforma desenvolvida majoritariamente em PHP, foi realizada em ambiente de servidor local, envolvendo ajustes na interface, na comunicação entre módulos e no fluxo de troca de dados, garantindo a estabilidade na incorporação da nova funcionalidade.

A etapa de Testes e Ajustes visaram verificar se o sistema desenvolvido está funcionando adequadamente. Assim, foram conduzidos testes com documentos previamente selecionados e perguntas elaboradas em diferentes níveis de dificuldade, para verificar a coerência e a relevância das respostas fornecidas. Nos casos de inconsistência ou limitações no desempenho, foram realizados os ajustes necessários para aprimorar o funcionamento do sistema.

Por fim, a avaliação de desempenho do sistema foi conduzida com base em métricas amplamente utilizadas na literatura sobre recuperação de informação e de processamento de linguagem natural. Para medir a qualidade de recuperação (*retrieval*) dos dados, empregou-se a métrica Precision@k, enquanto a avaliação da qualidade das respostas geradas, utilizou a métrica ROUGE-L. Nas duas métricas, os valores próximos de 1 (um) representam um melhor desempenho. Além disso, mensurou-se o tempo e a acurácia geral das respostas, considerando o número de respostas corretas fornecidas pelo modelo em relação ao total de perguntas avaliadas.

IV. RESULTADOS E DISCUSSÕES

Este capítulo apresenta discussão dos resultados obtidos com o desenvolvimento da metodologia proposta.

A. Revisão Sistemática de Literatura

A primeira etapa da Revisão Sistemática de Literatura consistiu na formulação da pergunta norteadora da pesquisa: “Quais são os principais modelos de Inteligência Artificial usados em sistemas de perguntas e respostas?”. A partir dos resultados obtidos através da pesquisa orientada pela pergunta foram selecionados os trabalhos relevantes a serem estudados.

Os estudos analisados apresentam aplicações de sistemas baseados em modelos de linguagem em diferentes áreas, como o setor jurídico [8], [9], educação [10], [11] e aprendizado de idiomas [12]. A maioria dos estudos empregou os modelos GPT-3.5 e GPT-4 devido ao alto desempenho e à facilidade de integração via API (Interface de Programação de Aplicação), embora o custo por tokens (custo por carga de processamento) seja uma limitação recorrente [8]. Dessa forma, como alternativa, os trabalhos recorreram a modelos de código aberto, como LLaMA 2, FLAN-T5 e Mistral-7B, que eliminam custos operacionais, mas exigem infraestrutura computacional mais robusta [13], [14].

A técnica RAG (*Retrieval-Augmented Generation*) aparece como estratégia central nos estudos revisados, pois permite dividir documentos em trechos menores e recuperar somente o conteúdo relevante para a pergunta. Assim, se reduz o texto encaminhado ao modelo significativamente, e consequentemente também se diminui o consumo de tokens nas APIs e a carga computacional ao executar modelos locais. As principais bibliotecas utilizadas são o LangChain e o LlamaIndex. O LangChain se destaca por sua flexibilidade e alto grau de customização, enquanto o LlamaIndex apresenta melhor desempenho em sistemas de perguntas e respostas específicos sobre documentos, apesar de menos flexível [15].

Alguns trabalhos também exploram otimizações para LLMs de código aberto, como o uso de QLoRA no Mistral-7B, que possibilita *fine-tuning* eficiente em GPUs com memória reduzida [14]. No campo do reconhecimento de voz, a biblioteca Whisper [12] se destaca por ter precisão em múltiplos idiomas, embora o custo proporcional ao tempo de áudio na API possa ser uma limitação.

Por fim, as análises realizadas nos estudos selecionados apontam para diferentes formas de avaliação dos sistemas. Muitos trabalhos utilizam precisão das respostas e tempo de resposta como critérios principais [12]. Além disso, também são encontradas avaliações subjetivas, com usuários atribuindo notas à qualidade e rapidez das respostas [10]. Contudo, outros estudos adotam métricas formais de classificação e recuperação de informação, como precisão, recall, F1-score, ROUGE e acurácia para mensurar a capacidade dos sistemas de encontrar, recuperar e gerar respostas corretas com base nos documentos [14], [16], [17].

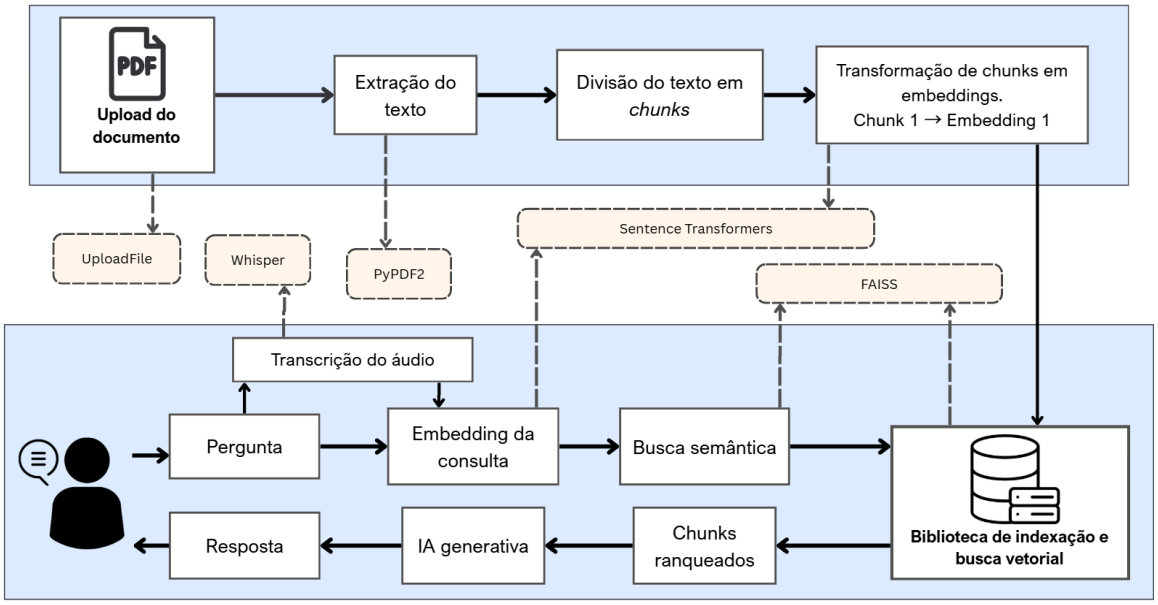

Figura 4: Etapas do Funcionamento do Sistema.

B. Análise e Seleção das Ferramentas

Nesta etapa o objetivo é avaliar as ferramentas identificadas na revisão e selecionar aquelas que melhor se adequaram aos requisitos do projeto. O primeiro passo foi decidir qual modelo de IA seria utilizado. Desse modo, optou-se por testar exclusivamente modelos de código aberto, uma vez que não apresentam custos de uso por token e possuem licenças permissivas para fins acadêmicos e comerciais. Os modelos avaliados foram OpenHermes, Falcon, Mistral e FLAN-T5, executados por meio da interface genérica text-generation-webui, que permite padronizar a comparação entre diferentes LLMs.

O modelo FLAN-T5 foi descartado porque suas respostas limitavam-se a copiar trechos literais do texto, demonstrando pouca habilidade inferencial. Entre os modelos avaliados, o Mistral-7B apresentou o melhor desempenho, com respostas mais coerentes, maior velocidade e menor ocorrência de erros gramaticais.

Após a escolha do modelo, foram definidas algumas das principais ferramentas do sistema. O *framework* selecionado para aplicação da API foi o FastAPI, devido ao suporte nativo nas operações assíncronas. Na implementação da técnica RAG (*Retrieval-Augmented Generation*), optou-se por uma construção manual, a fim de permitir maior controle sobre as etapas de extração, *chunking*, indexação e recuperação de trechos.

Dessa forma, para o banco vetorial, adotou-se o FAISS (*Facebook AI Similarity Search*), uma biblioteca otimizada de busca de vetores em alta dimensão, utilizada em sistemas de recuperação semântica. Por fim, para o módulo de comando por voz, utilizou-se o Whisper, modelo de reconhecimento de fala desenvolvido pela OpenAI, sendo responsável por transcrever o áudio enviado pelo usuário e produzir o texto, posteriormente interpretado pelo sistema.

C. Desenvolvimento do Sistema

O desenvolvimento do sistema foi orientado pelos resultados obtidos na revisão de literatura, e análise e seleção das ferramentas, que indicaram a eficácia do uso da técnica RAG (Geração Aumentada de Recuperação) em sistemas de perguntas e respostas baseados em texto.

A Figura 4 apresenta as etapas de funcionamento do sistema e as principais ferramentas utilizadas em cada uma delas. A primeira fase consiste na importação do documento PDF a ser consultado e na extração de seu texto por meio da biblioteca PyPDF2. Depois, o sistema segmenta esse texto em *chunks* e os converte em vetores (*embeddings*) utilizando a biblioteca Sentence Transformers. Esses *embeddings* são armazenados no FAISS, uma biblioteca de indexação e busca vetorial.

A segunda fase inicia quando o usuário formula uma pergunta, seja por texto ou por áudio. Assim, caso seja uma entrada de áudio, a transcrição é realizada pela API Whisper. A pergunta em formato de texto é convertida em vetor pela biblioteca Sentence Transformers. A partir disso, o FAISS compara o vetor da pergunta aos vetores dos *chunks* armazenados, identificando e ranqueando aqueles com maior similaridade semântica.

A Figura 5 ilustra um exemplo desse processo: observa-se que vetores associados às palavras “cão” e “cachorro” ocupam posições próximas no espaço vetorial, o que faz com que ambos sejam reconhecidos como semanticamente relacionados durante a busca.

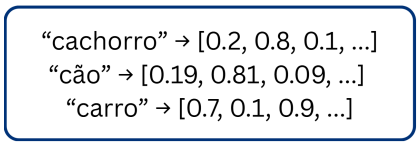

Figura 5: Exemplo de Vetores associados a palavras.

Os *chunks* mais relevantes são então enviados ao modelo de IA Mistral, responsável por interpretar o conteúdo recuperado e formular a resposta final. Contudo, é importante destacar que o modelo não cria informações que não estejam presentes nos *chunks* retornados; caso a informação não seja encontrada no documento, o sistema aponta essa ausência ao usuário.

D. Integração com ETC

O sistema foi desenvolvido para operar de forma integrada ao Editor de Texto Coletivo (ETC), uma plataforma de escrita coletiva desenvolvida majoritariamente em PHP e executada em um servidor Apache. A estrutura também hospeda a interface do usuário do projeto, sendo responsável por exibir botões de controle, histórico de consultas e gerenciamento dos documentos utilizados nas interações. Entretanto, devido à necessidade de aplicar modelos de Inteligência Artificial para o processamento de linguagem natural, foi necessário empregar um servidor adicional baseado em Python e executado com o Unicorn. Assim, o servidor permite o uso mais eficiente de bibliotecas específicas de IA e de processamento de voz.

A Figura 6 representa a arquitetura do sistema integrado ao ETC, composta por quatro módulos principais: Interface, API, RAG e Voice. A interface, localizada no front-end, é responsável pela interação com o usuário e pelo envio de solicitações. As solicitações são encaminhadas à API, desenvolvida com o framework FastAPI, que faz a ponte entre o sistema em PHP e os módulos em Python, executadas no servidor Unicorn. O Ollama, localizado no módulo RAG, é uma plataforma que permite executar modelos de linguagem (LLMs) localmente. No projeto, ele é responsável por carregar e gerenciar o modelo Mistral 7B-Instruct.

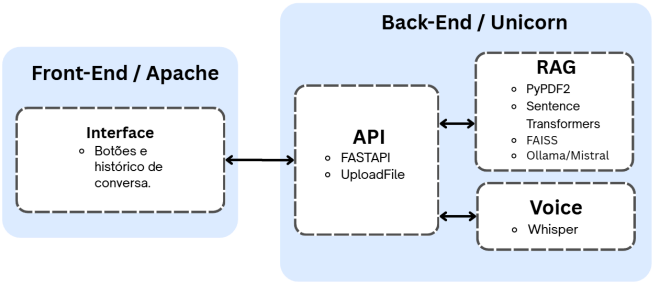

Figura 6: Arquitetura do Sistema.

A Figura 7 apresenta a página de edição de texto da plataforma ETC. No destaque (em vermelho), observa-se o botão que abre o Sistema de Consulta baseado em IA, cuja interface é mostrada na Figura 8.

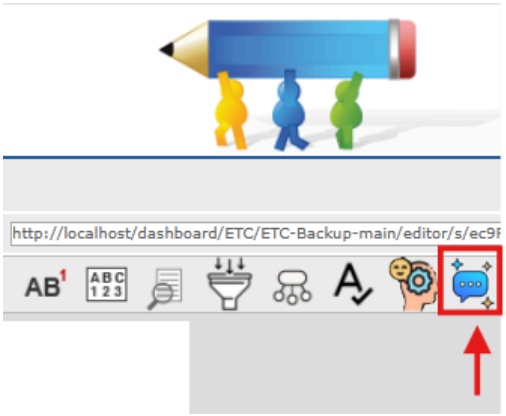

Figura 7: Interface do ETC com o Sistema de Consulta.

Na interface do Sistema de Consulta, apresentada na Figura 8, o usuário pode indexar novos documentos, remover documentos existentes, registrar e enviar perguntas, bem como visualizar, exportar e limpar o histórico de interação.

Além disso, caso algum documento seja removido da pasta de uploads, há também a opção de atualizar o índice para que apenas os arquivos válidos sejam considerados na recuperação de informações. A interface exibe ainda

Figura 8: Interface do Sistema de Consulta.

indicadores sobre o número de documentos indexados, a quantidade de perguntas realizadas na sessão, e a situação dos módulos de API e do sistema de comando de voz. No histórico exportado, o usuário consegue verificar se cada pergunta foi feita por voz ou texto, quais documentos foram utilizados como fonte para gerar a resposta, e a data e horário de cada interação.

E. Testes e Ajustes

Os testes de funcionamento foram conduzidos para avaliar a estabilidade da aplicação, a consistência da recuperação de documentos e o desempenho do módulo de comando de voz. Inicialmente, o sistema utilizava o ChromaDB como banco vetorial; entretanto, durante os testes observou-se instabilidade e dificuldades na recuperação de trechos longos. Por esse motivo, optou-se pela substituição do ChromaDB pelo FAISS, uma biblioteca desenvolvida pela Meta. O FAISS demonstrou maior velocidade, estabilidade e menor propensão a falhas durante a operação.

Para verificar o funcionamento geral da solução, foram realizados testes com múltiplos documentos PDF contendo estruturas e formatos distintos. O objetivo foi assegurar que o processo de extração textual, divisão em *chunks*, indexação e recuperação operasse corretamente em diferentes condições. Assim, parte desses testes foram realizados com o sistema já integrado ao ETC, para avaliar o comportamento da ferramenta dentro do fluxo real de uso. Os testes indicaram que o sistema manteve desempenho estável, exibindo corretamente o histórico de consultas, a lista de documentos indexados e as respostas geradas pelo modelo.

Além disso, também foi avaliado o módulo de comando de voz, responsável por transcrever perguntas realizadas oralmente. Nesta etapa, constatou-se que ruídos ambientais influenciam significativamente a qualidade das transcrições feitas pelo Whisper e para minimizar esses erros, utilizou-se o microfone Lark M2S sem fio com tecnologia de redução de ruído.

F. Avaliação de Desempenho

A avaliação de desempenho do sistema foi conduzida a partir de um conjunto de nove perguntas, divididas em três categorias:

- Perguntas factuais: respostas diretas, localizáveis no texto.
- Perguntas inferenciais: exigem interpretação, síntese ou relação entre trechos próximos.
- Perguntas multi-hop: dependem de duas ou mais partes distintas do documento.

A avaliação do sistema foi realizada a partir de um artigo científico, que descreve o desenvolvimento de uma funcionalidade de reconhecimento de expressões faciais integrada ao ETC [18].

As perguntas factuais envolveram a busca de informações de bibliotecas utilizadas, autoria e expressões faciais identificadas pela funcionalidade. As perguntas inferenciais exigiram que o sistema encontrasse a justificativa para decisões metodológicas e a relação entre aspectos técnicos e a experiência dos usuários. Por fim, as perguntas do tipo multi-hop demandaram a integração de informações provenientes de diferentes seções do documento, como os achados da Revisão Sistemática de Literatura, limitações apontadas pela literatura e sugestões de trabalhos futuros.

Para cada pergunta, o sistema executou o pipeline completo: busca semântica, seleção dos três melhores trechos (avaliadas pela Precision@3) e geração de resposta via modelo Mistral-7B. As respostas geradas foram avaliadas pelas métricas ROUGE-L e acurácia.

A métrica Precision@k [19] quantifica a proporção de trechos (*chunks*) relevantes entre os k primeiros resultados retornados pelo mecanismo de busca vetorial do RAG. Assim, essa métrica é empregada para avaliar a capacidade do sistema de recuperação, em priorizar os segmentos mais informativos e para responder à pergunta do usuário. A definição formal é:

$$P@K = \frac{TPk}{TPk + FPk}$$

Na fórmula, TP_K representa o número de trechos relevantes presentes entre os top-k resultados e FP_K representa o número de trechos irrelevantes entre esses mesmos k retornos. No presente estudo, utilizou-se k = 3, avaliando assim os três primeiros *chunks* recuperados para cada pergunta.

A métrica ROUGE-L avalia o grau de sobreposição de conteúdo entre a resposta produzida pelo sistema e a resposta de referência. O ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) é um conjunto de métricas amplamente utilizado em tarefas de sumarização automática [20]. A métrica permite avaliar similaridade semântica e preservação de estrutura textual, mesmo quando há pequenas variações lexicais ou sintáticas. Neste trabalho, o ROUGE-L foi calculado utilizando a biblioteca Rouge do Python.

Por fim, a Acurácia geral do sistema foi avaliada considerando a proporção de respostas corretas, avaliadas pelo autor, dentre o total de perguntas aplicadas: [21]:

$$Acurácia = \frac{Números\ de\ Respostas\ Corretas}{Total\ de\ Perguntas}$$

Por definição, a resposta foi considerada correta, quando transmitia a mesma informação essencial da resposta de referência, independentemente de variações redacionais. Dessa forma, esse cálculo permite avaliar a capacidade global do modelo em responder adequadamente às questões propostas.

As Perguntas Factuais obtiveram o melhor desempenho (Precision@3 = 0.56, ROUGE-L = 0.60), com todas as três perguntas respondidas corretamente. As respostas factuais estão presentes de forma explícita no documento, desse modo, o sistema conseguiu recuperar ao menos um ou dois trechos diretamente relevantes com as perguntas.

As Perguntas Inferenciais apresentaram o menor desempenho entre as categorias (Precision@3 = 0.11; ROUGE-L = 0.12). O sistema respondeu corretamente apenas 1 das 3 perguntas, o que contribuiu para reduzir o ROUGE-L médio. As duas perguntas que não foram respondidas dependiam de informações presentes em um mesmo trecho extenso do texto original, o que levantou a

hipótese inicial de que esse segmento pudesse ter sido fragmentado em múltiplos *chunks* durante a etapa de divisão. Após a verificação, constatou-se que o trecho relevante estava devidamente armazenado no banco vetorial. Assim, descartou-se o problema de segmentação e concluiu-se que a limitação está relacionada ao próprio mecanismo de busca do FAISS, que tende a ser penalizado ao recuperar *chunks* longos e semanticamente densos, contendo diversos assuntos mesclados. Logo, esse tipo de estrutura dificulta a identificação precisa do subtema específico solicitado na pergunta, reduzindo a capacidade do recuperador em priorizar o trecho mais relevante.

As Perguntas Multi-hop apresentaram desempenho intermediário (Precision@3 = 0.33; ROUGE-L = 0.40). O sistema respondeu corretamente às três perguntas e recuperou ao menos um *chunk* relevante em cada caso, o que explica a precisão de 0,33. O valor do ROUGE-L foi reduzido principalmente porque, em uma das respostas, o modelo gerou a informação em formato de tópicos, alterando a ordem e a continuidade das palavras em relação à resposta esperada. O ROUGE-L mede sobreposição sequencial, assim, essa mudança estrutural diminuiu a similaridade calculada pela métrica, mesmo que o conteúdo essencial estivesse correto.

O valor do ROUGE-L é diretamente influenciado pelos *chunks* recuperados, sua análise foi então feita separadamente. No total, o sistema respondeu corretamente sete das nove perguntas e, considerando apenas perguntas respondidas corretamente, o ROUGE-L apresentou média 0,48, indicando que, quando o modelo acerta, ele tende a captar a ideia central da resposta esperada, ainda que com variações de forma ou ordenação textual.

A tabela a seguir apresenta as médias gerais das métricas avaliadas, contemplando as nove perguntas feitas ao sistema:

Tabela 1: Resultados Gerais das Métricas Aplicadas.

Métrica	**Valor Médio**	**Interpretação**
Precision@3	0,33	33% dos trechos recuperados eram relevantes
ROUGE-L	0,38	38% de similaridade média com a resposta esperada
Acurácia	0,777	77,7% das respostas estavam corretas
Latência	2,9	Tempo médio de resposta em segundos

O valor da precisão indica que, de modo geral, as perguntas conseguiram recuperar ao menos um *chunk* relevante entre os três retornados. Neste sentido, esse resultado é satisfatório, considerando que algumas respostas estavam presentes apenas uma vez no documento, o que naturalmente reduz a probabilidade de o mecanismo de recuperação identificar múltiplos trechos relevantes para determinadas perguntas.

O tempo médio de resposta do sistema foi de 2,94 segundos. Esse tempo de resposta é esperado para sistemas RAG executados localmente, uma vez que a latência total depende principalmente da busca vetorial no FAISS e da geração da resposta pelo modelo de linguagem. Perguntas que exigiam raciocínio mais complexo ou respostas mais extensas apresentaram latências ligeiramente maiores, especialmente nas questões classificadas como multi-hop (3,95s em média).

A acurácia de 77,7% indica que o sistema foi capaz de responder corretamente à maioria das perguntas. No entanto, esse valor reflete o desempenho do sistema na totalidade, e não exclusivamente a capacidade do modelo Mistral. Isso evidencia que a principal limitação está na etapa de recuperação de informações (RAG), sugerindo a necessidade de aprimorar o mecanismo de busca e seleção de *chunks* para que o modelo possa gerar respostas ainda mais consistentes.

Por fim, para efeito de comparação com trabalhos semelhantes, destaca-se o estudo de [14], que também empregou métricas automáticas para avaliar respostas geradas por modelos de linguagem. A autora utilizou a métrica ROUGE para mensurar a similaridade entre as respostas produzidas pelo sistema e uma resposta de referência definida pelo pesquisador. Apesar de possíveis diferenças na implementação, por existirem variações como ROUGE-1, ROUGE-2 ou ROUGE-L, trata-se da mesma família de métricas. Dessa forma, os valores obtidos neste projeto são comparáveis aos resultados apresentados em [14], da qual obtiveram valores variando de 0,27 à 0,74 em diferentes modelos de LLM. Portanto, conclui-se que ambos os sistemas atingiram um nível similar de retenção de conteúdo relevante nas respostas.

V. CONCLUSÕES

O presente trabalho apresentou o desenvolvimento, a integração e a avaliação de um sistema de consulta inteligente baseado na técnica Retrieval-Augmented Generation (RAG), incorporado ao Editor de Texto Coletivo (ETC). A solução proposta permite que usuários realizem perguntas em linguagem natural sobre documentos previamente indexados, utilizando comandos de texto e de voz. O sistema foi desenvolvido com modelos e ferramentas de código aberto, destacando o uso do modelo Mistral-7B-Instruct via Ollama, o mecanismo de recuperação vetorial FAISS e o Whisper para transcrição de áudio. Os resultados demonstraram que o sistema foi capaz de responder corretamente 77,7% das perguntas formuladas, evidenciando sua viabilidade para apoiar o processo de escrita e consulta de informações no ETC.

A análise das métricas de desempenho mostrou, porém, que a principal limitação reside na etapa de recuperação dos trechos relevantes: quando o sistema recupera bons trechos, o modelo Mistral tende a formular respostas semanticamente adequadas; quando a recuperação falha, o desempenho global se reduz. Assim, conclui-se que o desempenho limitado não se deve ao modelo Mistral, mas principalmente ao mecanismo de recuperação semântica. Isso indica que futuras melhorias devem se concentrar no processo de *chunking* e no método de indexação. A redução do tamanho dos *chunks*, por exemplo, pode melhorar a precisão do FAISS, reduzindo a perda de informações importantes dentro de trechos muito grandes. Além disso,

outra melhoria possível seria o uso de *embeddings* mais especializados para textos longos ou inclusão de mecanismos de filtragem semântica secundária antes da geração.

Do ponto de vista pedagógico, o sistema contribui para o uso ético da informação ao manter a rastreabilidade entre as respostas geradas e os documentos que as fundamentam. Essa característica possibilita a mediação docente no uso da ferramenta, orientando os estudantes quanto à formulação de perguntas, à avaliação da qualidade das fontes e à interpretação crítica das respostas fornecidas pela IA. Em cenários educacionais, a solução pode ser explorada em atividades como elaboração de resumos críticos, revisões de literatura colaborativas e debates fundamentados em textos, fortalecendo práticas de autoria e reflexão acadêmica no ambiente do ETC.

Desse modo, a possibilidade de ampliar a ferramenta para interpretar textos presentes em imagens pela tecnologia OCR (Reconhecimento Óptico de Caracteres), funcionalidade atualmente não suportada pelo Mistral, pode expandir a aplicabilidade em contextos educacionais.

Por fim, o sistema desenvolvido cumpriu seu objetivo de integrar uma funcionalidade de consulta inteligente ao ETC, oferecendo apoio ao processo de escrita e possibilitando, a partir de tecnologias baseadas em IA, qualificar a produção textual desses usuários. Os resultados obtidos demonstram o potencial da abordagem RAG em ambientes educacionais e indicam caminhos para aprimoramentos futuros, especialmente na etapa de recuperação, que se mostra o elemento mais determinante para elevar o desempenho geral da solução.

VI. REFERÊNCIAS

[1] L. Oliveira e M. Pinto, A inteligência artificial na educação: ameaças e oportunidades para o ensino-aprendizagem. Porto: Escola Superior de Media Artes e Design, 2023. Disponível em: https://edicoes.ipp.pt/index.php/books/catalog/book/99 . Acesso em: 17 jun. 2025.

[2] P. P. Ray, “ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope,” Internet of Things and Cyber-Physical Systems, vol. 3, pp. 121–154, 2023. DOI: https://doi.org/10.1016/j.iotcps.2023.04.003 . Acesso em: 26 mar. 2025.

[3] F. P. Navarro, Uso da inteligência artificial para recuperação da informação com abordagem semântica: modelo de aplicação para documentos textuais em ambientes digitais. 2021. Tese (Doutorado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Marília, 2021. Disponível em: https://repositorio.unesp.br/handle/11449/204693 . Acesso em: 6 maio 2025.

[4] D. Jurafsky e J. H. Martin, Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Draft da 3. ed., 2023. Disponível em: https://web.stanford.edu/~jurafsky/slp3/ . Acesso em: 1 jun. 2025.

[5] P. A. Behar, Recomendação pedagógica em educação a distância. Porto Alegre: Penso, 2019. Ebook. ISBN 9788584291588. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788584291588.

[6] D. Gough, S. Oliver e J. Thomas, An introduction to systematic reviews. Londres: Sage, 2012.

[7] F. J. Paz e S. C. Cazella, “Integrando sistemas de recomendação com mineração de dados educacionais e learning analytics: uma revisão sistemática da literatura,” Novas Tecnologias na Educação, v. 16, n. 1, jul. 2018.

[8] D. F. Pereira, Integração de LLM e RAG para análise de documentos jurídicos. Trabalho de Conclusão de Curso – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2024.

[9] L. S. Barbosa, Um chatbot especializado para o contexto da Universidade Federal de Ouro Preto. Monografia – Universidade Federal de Ouro Preto, Ouro Preto, 2023.

[10] A. Moraes, Assistente de ensino baseado em um modelo de linguagem de grande escala brasileiro. Trabalho de Conclusão de Curso – Universidade Federal do Pará, Cametá, 2024.

[11] B. Zheng, Chatbots baseados em Inteligência Artificial: um protótipo para uma instituição de ensino superior. Dissertação (Mestrado) – Instituto Superior de Contabilidade e Administração de Coimbra, Coimbra, 2024.

[12] G. D. Ribeiro, Aplicação de IA Generativa na Prática de Línguas: Criando um Chatbot Conversacional Multimodal com OpenAI API. Trabalho de Conclusão de Curso – Universidade Estadual Paulista, Bauru, 2024.

[13] R. S. Resende, Criação de um chatbot para responder dúvidas sobre editais de concursos com processamento de linguagem natural e Python. Trabalho de Conclusão de Curso – Instituto Federal do Espírito Santo, Cachoeiro de Itapemirim, 2024.

[14] O. Karaim, Application of LLMs for a Chatbot System in the Logistics Industry. Trabalho de Conclusão de Curso – Ukrainian Catholic University, Lviv, 2024. Acesso em: 29 jun. 2025.

[15] Tamanna, “LangChain vs. LlamaIndex: A Comprehensive Comparison for Retrieval-Augmented Generation (RAG),” Medium, 24 out. 2024. Disponível em: https://medium.com/@tam.tamanna18/langchain-vs-llamaindex-a-comprehensive-comparison-for-retrieval-augmented-generation-rag-0adc119363fe . Acesso em: 20 mai. 2025.

[16] T. A. A. Medeiros, Transformação digital no setor automotivo: desenvolvimento de um chatbot com modelos de linguagem para a extração de conhecimento de manuais automotivos. Trabalho de Conclusão de Curso – Universidade Federal do Rio Grande do Norte, Natal, 2023.

[17] K. E. Rajakumari, G. Thenkanishankar, J. D. Syiem, S. Lavanya e S. Jothi, “AI doc question and answering system,” International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI), 2025.

[18] A. G. Alves, M. A. R. Guizzo, A. Lorandi e P. A. Behar, Sistema de identificação de expressões faciais básicas em um Editor de Texto Coletivo. Revista Ibérica de Sistemas e Tecnologias de Informação, n. 54, p. 88–103, jun. 2024. Disponível em: https://www.risti.xyz/issues/risti54.pdf . Acesso em: 6 maio 2025.

[19] A. Ammar, A. Koubaa, O. Nacar e W. Boulila, “Optimizing Retrieval-Augmented Generation: Analysis of Hyperparameter Impact on Performance and Efficiency,” arXiv, 2025. Disponível em: https://arxiv.org/abs/2505.08445 .

[20] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” Information Sciences Institute, University of Southern California, Marina del Rey, 2004.

[21] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu e Z. Liu, “Evaluation of Retrieval-Augmented Generation: A Survey,” arXiv, 2024. Disponível em: https://arxiv.org/abs/2405.07437 .